

A TCP/IP example

the **TCP/IP** communication **protocols** in retrieving a Web page from a server.

The user interacts with programs at the **application layer**. In this case, we are using a Web **client** program to communicate with a Web **server** program. The user has entered a **URL** in order to retrieve a Web page from the server at *som.csudh.edu*.

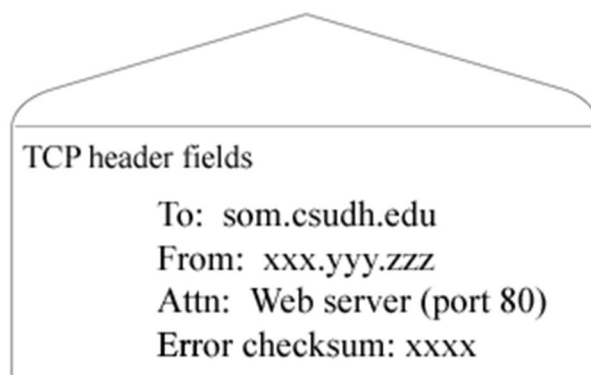
When the user hits the Enter key, the application program simply passes a message saying:

Get :
som.csudh.edu/fac/lpress

to the **TCP** program at the **transport layer**. TCP is responsible for error-free delivery of the message to the destination **host**. (It will let the application program know if it cannot deliver the message for some reason).



The TCP program adds **header** information to the message. You can think of it as putting the message in an envelope. The header (envelope) has various fields, including the address of the source and destination hosts and an error **checksum**. Since there may be many server programs running on the destination host, it also indicates that this message is intended for the Web server. The Web server is indicated by the **port** number, 80. (Other server programs would be assigned other port numbers).

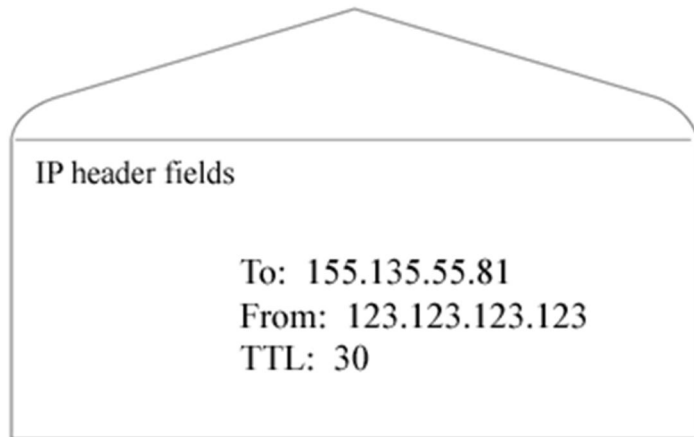


The TCP program then passes the message plus its new header (a **TCP segment**) to the **IP** program at the **network layer**.

The IP program adds another header. You can think of it as putting the transport layer envelope inside a larger envelope. The IP header contains only the information the routers need to deliver the **packet** to the destination host. The IP header fields include the source and destination IP addresses and the **time-to-live (TTL)**.

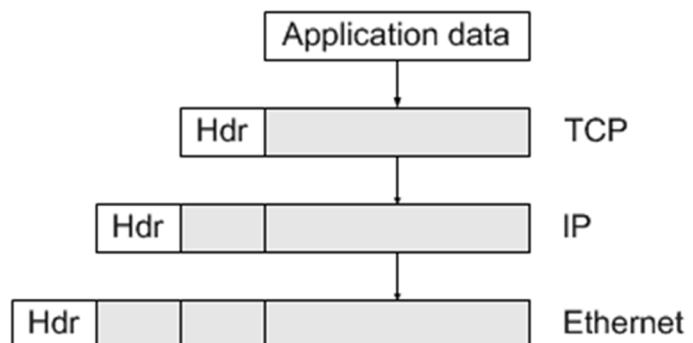
The IP "envelope" is referred to as a **packet** or a **datagram**.

Note that TCP has converted the **domain names** *som.csudh.edu* and *xxx.yyy.zz* to **IP addresses**. It looked the IP addresses up using a system of **domain name servers**. The TTL field is the number of router hops the packet should go through before giving up and deciding that it cannot be delivered. Each router that receives the packet will reduce the TTL field by one, and, if it reaches 0, the packet is discarded.



The packets are moved within the LAN -- from the sending host to the router -- by the hardware, software and firmware at the **data link** and **physical** layers at the bottom of the protocol **stack**.

If we assume we have an Ethernet LAN, **encapsulation** can be viewed as shown here. Each program in the stack adds its own header. They are



removed (the "envelopes" are opened) at the receiving end.	
--	--

In studying this example, note that the programs at each **layer** operate independently. An application program does not worry about how TCP/IP delivers the message, how many routers it goes through, where the destination host is located, what operating system it is running, whether or not there are any wireless links along the route, etc. Each layer hides the exact way it works from its "boss" above -- it just gets the job done.

Note also that the TCP/IP software only moves data. Messages requesting Web pages are handled the same way as messages containing Web pages or e-mail or video or sound. The routers only look at the IP packet headers to determine the best way to move the packet toward its destination.

This has two benefits: it enables the routers to process packets rapidly and it keeps the application intelligence at the edges of the network. The latter is subtle, but extremely important since innovation (new applications) are developed by users, not the network operators. The network is fast but **dumb** -- it merely passes bits.